Association Rule Mining

استخراج قواعد انجمنی

ارائه دهنده: سیدحامد میرلوحی

- Frequent Patterns Mining
- Frequent Itemsets Mining

Association Rules

Pattern Mining

- Pattern : یک سری چیزهای تکرار شونده را به عنوان الگو می شناسیم.
- Pattern Mining یک مفهوم کلی است. چرا؟

Classification

- الگوی ما یک classifier ای است که ما آن را کشف می کنیم

Clustering

- الگویی از جنس شباهت است

Regression

- توصیف ریاضی یک الگو

انواع الگوهای تکرار شونده:

- پیچیدگی ساختار

- Item set : در این نوع، ترکیب مهم است و لی ترتیب و توالی مهم نیست
  مثال: اقلامی که به صورت مکرر از یک فروشگاه خریداری می شوند.

- Subsequence : علاوه بر ترکیب، ترتیب نیز در آنها مهم است

  کاربرد تجاری و اقتصادی آن زیاد است. مثال: پولشویی

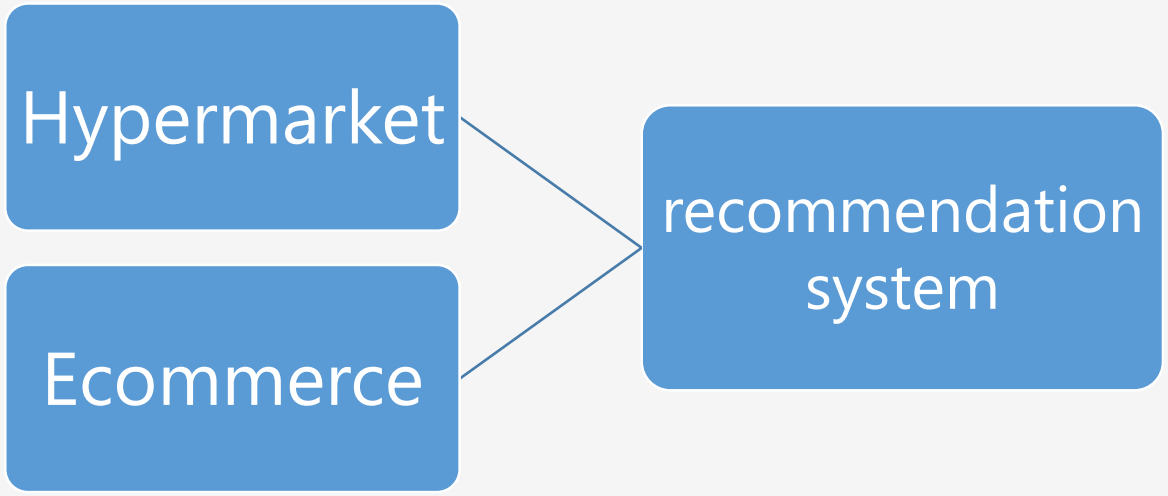
- Substructure : ساختاری پیچیده تر از زیردنباله دارد. ترتیب در آنها از جنس سلسله مراتبی است. برعکس زیردنباله که با ساختار ترتیبی نشان داده می شوند اغلب به شکل درخت نشان داده می شوند. مثال: ویروس کامپیوتری

مثالی از کاربرد تحلیل الگوهای خطی

- Shopping Cart Analysis
- Market Basket Analysis

- کاربرد: می توان با استفاده از این تحلیل سیستم های پیشنهاد گرا ارائه داد.

سایر مثال ها	موتور جستجوی گوگل
	نسخ بدخط پزشکان در داروخانه ها



تحلیل سبد خرید

TID	Item
1	نان، پنیر، شیر
2	نان، پنیر، کره، شیر
3	نان، شکر، شیر
4	شکر، آرد، شیر

به دو صورت این الگو را می توان بیان کرد:

شیر ⟶ **نان**

نان ⟶ **شیر**

نحوه نمایش ریاضی یک Rule :

A ⟶ **B**

۲ ویژگی مهم برای محاسبه قوانین وابستگی

- ۱) Support: (پشتیبان) : تعداد دفعات وقوع یک الگو را بیان می کند. (Usefulness)
- ۲) Confidence: (اطمینان): قابلیت اطمینان این الگو را بیان می کند. (Certainty)

- هر دو ویژگی بر حسب درصد بیان می شوند:
- مثال قبل:

شیر ⟶ نان **sup:75%** (3/4) **conf:100%** (3/3)

نان ⟶ شیر **sup:75%** (3/4) **conf:75%** (3/4)

۲ مفهوم مهم دیگر

- MST : minimum support threshold
- MCT: minimum confidence threshold

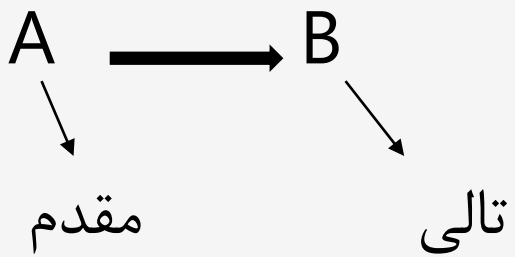
- Strong Rule: قاعده ای است که Support آن بیش از MST و Confidence آن بیش از MCT باشد.

توصیف به زبان ریاضی

Items $I: \{I_1 , I_2 , \dots , I_n\}$

Database $D:\{T_1 , T_2 , \dots T_n\}$

$T_i \subset I$



تعریف یک Rule :

$A \subset I \quad , \quad B \subset I$

$A \neq \emptyset \quad , \quad B \neq \emptyset$

$A \cap B = \emptyset$

توصیف به زبان ریاضی

- بیان Support یک Rule :

- Support(A -> B) = P (A ∩ B)

Support(A -> B) = Support(B -> A)

- بیان Confidence یک Rule :

Confidence (A -> B) = P (B | A)= $\frac{Support(A \cap B)}{Support(A)}$

Rule Reduction

- می خواهیم با کمترین Rule، بهترین توصیف را از کل سیستم و عملکرد آن داشته باشیم.

$$Conf(\text{نان} \rightarrow \text{شیر}) = \frac{\frac{3}{4}}{\frac{3}{4}} = 1$$

$$Conf(\text{شیر} \rightarrow \text{نان}) = \frac{\frac{3}{4}}{\frac{4}{4}} = \frac{3}{4}$$

کدام Rule را انتخاب کنیم

محاسبه تعداد رول های یک سیستم

فرض کنیم مجموعه آیتم ها برابر :

- Items I: $\{I_1 , I_2 , \dots , I_m\}$

۱-item set : $\{I_1\} , \{I_2\} , \dots , \{I_M\}$

مجموعه های **1** موردی
تعداد: **M** تا

...

k-item set : {...} , ...

مجموعه هایی شامل **k** موردی
تعداد $\binom{m}{k}$ تا

...

M-item set: {یک مجموعه ام عضوی}

یک مجموعه **M** عضوی

محاسبه تعداد رول های یک سیستم

- جمع 1-itemset ها تا k-itemset ها می شود:

$$\sum_{k=1}^{m} \binom{m}{k} = 2^m - 1$$

این نشان دهنده این است که تعداد Rule های یک سیستم از مرتبه نمایی است

مثلا اگر تعدا اقلام یک فروشگاه را فقط ۱۰۰ قلم جنس فرض کنیم یعنی m=100 باشد آنگاه تعداد Rule هایی که قرار است بررسی شوند برابر $2^{100} - 1$ است.

۲ مفهوم مهم دیگر:

- ۱- **الگوی بسته (Close Item set)** : X یک مجموعه بسته است اگر Y وجود نداشته باشد که $X \subset Y$ و $SUPP(X) = SUPP(Y)$

یعنی X بسته است اگر با اضافه شدن یک مورد به آن Support آن حتما افت کند

یعنی خرید یا سفارشی که اگر یک مورد به آن اضافه شود حتما احتمال وقوع آن پایین بیاید

۲ مفهوم مهم دیگر:

- ۱- **الگوی بیشینه (Maximal Item set)** : X ماکسیمال است اگر هیچ Y را نتوان یافت که $X \subset Y$ و Y هم یک مورد رایج(متداول) باشد.

یعنی X ماکسیمال است اگر با اضافه شدن یک مورد به آن، دیگر از رایج بودن خارج شود و به یک سفارش خاص تبدیل شود.

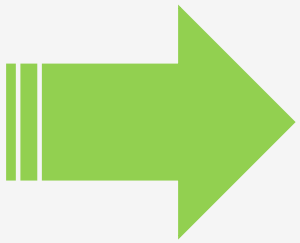
در مورد مجموعه بسته می گوییم، با اضافه شدن یک آیتم، ساپورت آن افت کند ولی در مورد مجموعه بیشینه می گوئیم با اضافه شدن یک آیتم نه تنها ساپورت آن افت کند، بلکه آن سفارش را از یک سفارش رایج به یک سفارش خاص تبدیل کند.

رابطه مجموعه های ماکسیمال و بسته

- Maximal + Frequent Pattern

SUPP > MST

- Closed + Frequent Pattern



هر الگوی ماکسیمال یک الگوی بسته است

A Priori Algorithm

- اصول کلی الگوریتم A priori :
- C_1 → L_1 → C_2 → L_2 → ...

- C : کاندیداها
- L : الگوهی نهایی و رایج
- → :Prune فاز = فاز تبدیل کاندید به الگوی نهایی
- → :Join فاز =فاز تبدیل الگوهای رایج به کاندیداها
- شرط خاتمه فرایند: $L_k = \emptyset$
- $L = L_1 \cup L_2 \cup \dots \cup L_{K-1} \cup L_K$
- حال از دل این مجموعه باید Rule ها را استخراج کنیم

The Apriori Algorithm

C_k: Candidate itemset of size k

L_k : frequent itemset of size k

L_1 = {frequent items};

for (k = 1; L_k !=∅; k++) **do begin**

C_{k+1} = candidates generated from L_k;

for each transaction t in database do

increment the count of all candidates in C_{k+1} that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

Implementation of Apriori

- **Step 1**

- C_1:??
- L_1:??

- **Step 2:**

- K = 1

Implementation of A priori

- **Step 3:**
  - ایجاد کاندیداهای مرحله $k+1$ از ترکیب اعضای L_k (فاز joining):

- $C_{k+1} = \{ A \cup \{b\} \mid A \in L_k , b \in \cup_k L_k , b \notin A\}$

$K+1$	k	۱
عضو	عضو	عضو

==

A اجتماعش با یک عضو تنها مثل b که A عضوی است از L های مرحله قبل و b عضوی است از اجتماع L های مرحله قبل و ضمنا b داخل A نیست

Implementation of A priori

- **Step 4:**
  - به ازای هر تراکنش مانند t، یک واحد(امتیاز) به تعداد حضور همه کاندیداهایی که در C_{k+1} هستند و در سفارش t نیز مشاهده می شوند اضافه می کنیم.
  - مثال: فرض کنیم در مرحله دوم باشیم، آنگاه:

TID	Item
1	نان، پنیر، شیر
2	نان، پنیر، کره، شیر
3	نان، شکر، شیر
4	شکر، آرد، شیر

Itemset	Count
نان، شیر	3
نان، پنیر	2
نان، آرد	0

Implementation of A priori

- **Step 5:**

- فاز Pruning:

$L_{k+1} = \{ C \mid C \in C_{K+1} , Count(c) \geq MST \}$

L_{k+1} مجموعه اعضای C_{k+1} است که تعداد حضور آن ها حداقل برابر MST باشد.

- **Step 6:**
- $\boldsymbol{k = k + 1}$

یک واحد به شمارنده اضافه می کنیم

Implementation of A priori

- **Step 7:**
  - اگر L_k یک مجموعه تهی بود، آنگاه به مرحله بعد می رویم و در غیر اینصورت به مرحله ۳ می رویم.
- **Step 8:**
- **$L = \cancel{L_1} \cup L_2 \cup L_3 \cup \ldots \cup L_K$**
  - مجموعه کلیه الگو ها را ایجاد می کنیم
- **Step 9:**
  - قواعد قوی را از بررسی اعضای L استخراج می کنیم و به عنوان خروجی بر می گردانیم

مثال:

MST = 0.5

$= {}^{2}/_{4}$

MCT = .7

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan

C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

نقطه قوت و ضعف الگوریتم

- + :
- پیاده سازی آن کار سختی نیست

- - :

- اتلاف منابع محاسباتی: تعداد Rule های خیلی زیادی را تولید میکند و دور می ریزد و همچنین حجم محاسیاتی سنگین در فاز ۴

نحوه استخراج Rule از یک Itemset

۱. ابتدا تمام زیر مجموعه های غیر تهی **l** را استخراج می کنیم.

۲. به ازای هر **l** است را بررسی می **A – l = B** که در آن **A --> B** ، قاعده **A ⊂** کنیم.(**partitioning**)

۳. در صورتی که **A --> B** قوی باشد، آنگاه به مجموعه قواعد اضافه می شود.

ادامه مثال قبل:

- L_2 = { {A, C} , {B, C} , {B, E} , {C, E} }

2 2 2 2

- L_3 = { {B, C, E} }

6

تعداد **Rule** های مورد بررسی: $(2^n - 2)$
2+2+2+2+6 = 14 رول

- L = $L_2 \cup L_3$

- L={ {A, C} , {B, C} , {B, E} , {C, E} , {B, C, E} }

ادامه مثال قبل:

	Rule	Confidence
{A , C}	A ⟶ C	$^2/_2$
	C ⟶ A	$^2/_3$
{B , C}	B ⟶ C	$^2/_3$
	C ⟶ B	$^2/_3$
{B , E}	B ⟶ E	$^3/_3$
	E ⟶ B	$^3/_3$
{C , E}	C ⟶ E	$^2/_3$
	E ⟶ C	$^2/_3$

	Rule	Confidence
{B , C , E}	B, C ⟶ E	$^2/_2$
	B, E ⟶ C	$^2/_3$
	C, E ⟶ B	$^2/_2$
	$E \longrightarrow B, C$	$^2/_3$
	$C \longrightarrow B, E$	$^2/_3$
	$B \longrightarrow C, E$	$^2/_3$

MCT = %70

نحوه نمایش داده ها:

- نحوه نمایش داده ها به صورت افقی:

T_{id}	$Item_s$
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

- نحوه نمایش داده ها به صورت عمودی:

Item	TID_s
A	10, 30
B	20, 30, 40
C	10, 20, 30
D	10
E	20, 30, 40

معرفی چند پارامتر ارزیابی مهم دیگر

- ✓Support
- ✓Confidence
- ➢Completeness
- ➢Lift
- ➢Discriminability
- ➢Leverage
- ➢Conviction

Completeness

Support(A---->B) = P (A ∩ B) = $Count\ (A \cap B)/_{n}$

Confidence(A---->B) = P(B | A) = $Count\ (A \cap B)/_{Count\ (A)}$

- **Completeness(**A---->B**) = P(A | B)** $= \dfrac{Count\ (A \cap B)}{Count(B)}$

آیا هر رول قوی، یک رول صحیح سودمند است؟

مثال: **در یک فروشگاه محصولات دیجیتال:**

- ✓تعداد سفارش ها: **10,000**
- ✓تعداد سفارش های داری video: **7,500**
- ✓تعداد سفارش های دارای game: **6,000**
- ✓تعداد سفارش های دارای video و game به صورت همزمان:**4,000**

• Rule مورد بررسی: خرید همزمان video و game ؟ game ----> video

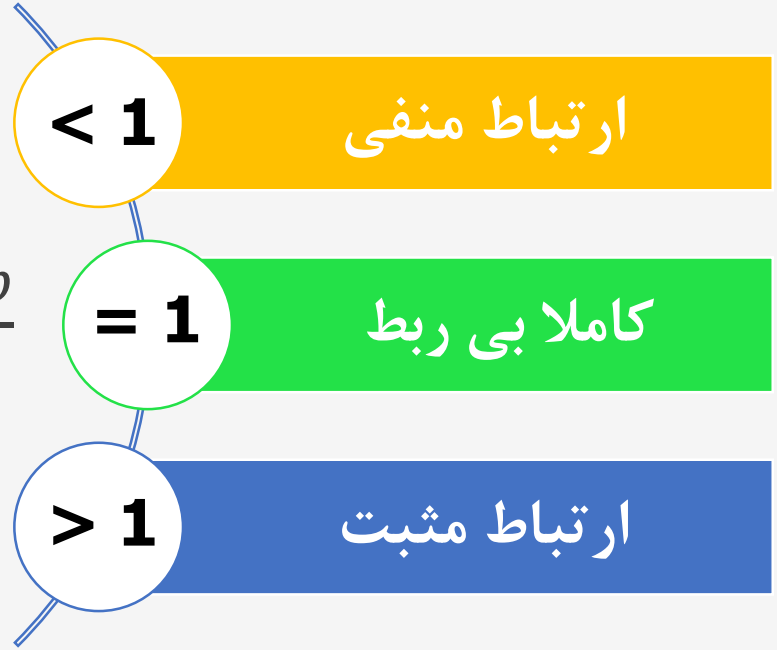
• Support=$^{4000}/_{10000}$ = 40% confidence=$^{4000}/_{6000}$ = 66.67%

اما الزاما این Rule قابل توجهی نیست. چرا؟؟؟

ازنظر آماری : **P(game) = 60%** **P(video) = 75%**

Lift

$$\text{Lift (A----->B)} = \frac{P(A \cap B)}{P(A) \,.\, P(B)} = \frac{conf}{P(B)} = \frac{comp}{P(A)}$$



پس قواعدی را در نظر می گیریم که Lift آنها بزرگتر از ۱ باشد

مثال:

- در مثال قبل:

- Completeness(A---->B) $= \frac{Count\ (A \cap B)}{Count(B)} = {}^{4000}/_{7500} = 53.33\%$

Lift(A---->B) $= \frac{P(A \cap B)}{P(A) \,.\, P(B)} = \frac{{}^{4000}/_{10000}}{{}^{7500}/_{10000} * {}^{6000}/_{10000}} = .88 \quad < 1$

Discriminability

- Disc(A---->B) = 1 - $\frac{Supp(A) - Supp(A \cap B)}{n_{total} - Supp(B)}$

- ضریب تفکیک پذیری یک Rule است.
- اگر A----> B به عنوان یک اصل باشد، (یعنی A همیشه همراه B است) آنگاه :

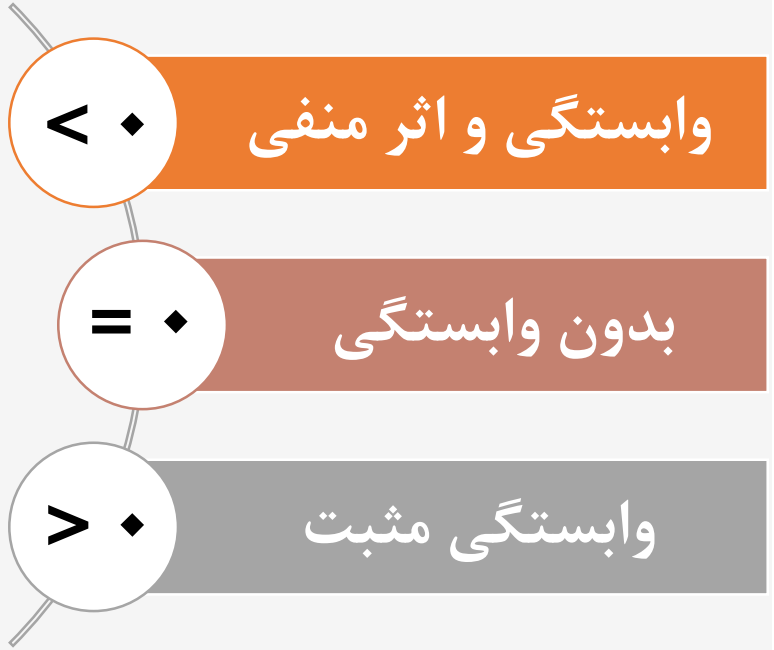
Support(A) = Support (A ∩ B)

در نتیجه:

Disc(A---->B) = 1 – 0 = 1

Leverage

- Leverage (A---->B) = $P(A \cap B) - P(A) * P(B)$



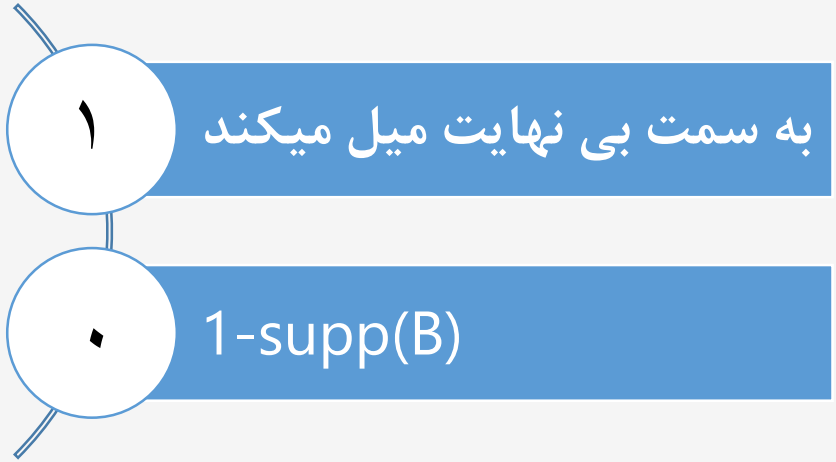
- وسعت پذیری یک Rule را نشان می دهد و مترادف با Lift است.

Conviction

- Conviction(A---->B) = $\frac{1 - Supp(B)}{1 - conf\ \ (A \rightarrow B)}$

- هر چه این پارامتر عدد بزرگتری را نشان دهد و به سمت بی نهایت میل کند رابطه بین A و B را به صورت بهتری بیان می کند و هرچه به سمت صفر میل کند این Rule ارزش چندانی ندارد.

- $conf\ \ (A \rightarrow B)$



مهمترین پارامترهای ارزیابی یک Rule

Support	کاربردی بودن
Confidence	میزان قابلیت اطمینان
Lift	منطقی بودن